

Undecidability of satisfiability in the algebra of finite binary relations with union, composition, and difference

Tony Tan, Jan Van den Bussche, and Xiaowang Zhang
Hasselt University

June 3, 2014

Abstract

We consider expressions built up from binary relation names using the operators union, composition, and set difference. We show that it is undecidable to test whether a given such expression e is finitely satisfiable, i.e., whether there exist finite binary relations that can be substituted for the relation names so that e evaluates to a nonempty result. This result already holds in restriction to expressions that mention just a single relation name, and where the difference operator can be nested at most once.

1 Introduction

The calculus (or algebra) of binary relations was invented by Peirce and Schröder and further developed by Tarski and his collaborators [Tar41, Pra92, Mad91]. Hence we will denote it by TA (for Tarski Algebra). TA consists of the operators union, complement, composition, and inverse, and provides the empty and the identity relations as constants. At present, this algebra (often extended with the transitive closure operator) provides a nice theoretical foundation for query languages for graph databases modeled as finite binary relational structures [SSVG93, tCM07, FGL⁺11, Woo12, LMV13]. Also practical graph database query languages such as Gremlin fit in this framework.

Specifically, given a vocabulary Γ of binary relation names, we can consider expressions built up using the names in Γ and the constants and operators mentioned above. These expressions serve as abstractions of query expressions evaluated on graph databases, viewed as relational structures over Γ . The result of a query is again a binary relation. For example, for $a, b \in \Gamma$, the expression $aaa - b$ asks for all pairs (x, y) such that one can walk from x to y in three steps using a -edges, but there is no direct b -edge from x to y . Here, the operation of composition is denoted simply by juxtaposition, and $-$ (set difference) can be expressed in terms of union and complement by $r - s = (r^c \cup s)^c$.

In this manner, one can express precisely the binary queries definable in FO^3 , the fragment of first-order logic with three variables [TG87, MV97]. In particular, one can translate effectively from an FO^3 formula with two free variables to a TA expression, and back. This connection with first-order logic provides immediate insight in the classical decision problem in the context of TA: given a vocabulary Γ and a TA expression e over Γ , is e satisfiable? That is, does there exist a structure I over Γ such that e on I evaluates to a nonempty result? Since satisfiability for FO^3 is undecidable [BGG97, Sch79], satisfiability for TA is undecidable as well.

This undecidability result can be sharpened considerably: it already holds for the fragment of TA consisting only of union, complement, and composition [AGN97]. In this paper, we show that undecidability continues to hold when we have only the relative form of complement provided by the set difference operation. Concretely, we consider a fragment of TA that we call the Downward Algebra (DA): its only operators are union, intersection, composition, and set difference. The name of this fragment is inspired by its salient property that, when viewing binary relations as directed graphs, DA expressions can only talk about pairs of elements formed by following edges in the forward (or downward) direction.¹ The focus on set difference, as opposed to general complement, is motivated by the database query language setting, where set difference is the standard form of negation [AHV95]. We will actually show that undecidability already holds for DA expressions in which the nesting depth of difference operators is at most two, and that use only a single relation name. We denote this fragment of DA by DA_2^1 .

Our result is also relevant to expressive description logics and dynamic logics. Indeed, DA expressions can be viewed as extended ‘role’ expressions in description logic, or ‘programs’ in dynamic logic [BCM⁺03, HKT00], so our result shows that satisfiability of such extended expressions or formulas is undecidable already for DA_2^1 . Known undecidability results for expressive dynamic/description logics assume either the full complement or the transitive closure operator [KRV14]. An undecidability proof given by Lutz and Walther [LW05] also uses only set difference on binary relations, but additionally needs the identity relation and the ‘diamond’ operator $\langle r \rangle = \{(x, x) \mid \exists y : (x, y) \in r\}$ on binary relations. On the other hand, dynamic logic where complement can be applied only relative to the identity relation (so-called ‘formula negation’), as well as to relation names (so-called ‘atomic negation’), is still decidable [LW05]. Thus, our result sharpens known undecidability results and helps delineating the boundary of undecidability. We repeat that DA contains neither the identity relation nor the diamond operator.

We should make clear that our result is specifically about satisfiability by a *finite* structure. The problem of deciding unrestricted satisfiability for DA expressions remains open.

This paper is further organized as follows. Section 2 defines DA, the frag-

¹A similar terminology has been used in the context of XPath, which is a form of TA used on tree structures as opposed to general graphs [Fig12].

ment DA_2^1 , and the corresponding satisfiability problem formally. Section 3 proves undecidability of finite satisfiability for general vocabularies. Section 4 reduces the problem to a vocabulary with just a single relation name. Section 5 concludes.

2 Satisfiability of DA expressions

Let Γ denote a finite vocabulary of binary relation names. The expressions e of DA over Γ are defined by the following grammar, where a ranges over the elements of Γ :

$$e ::= a \mid e \cup e \mid e \cap e \mid e - e \mid e \cdot e$$

The dot operator, which will denote composition, is often omitted when writing expressions, thus denoting composition simply by juxtaposition. For example, for $a, b \in \Gamma$, the expression $(a \cdot a - b) \cdot a$ is also written as $(aa - b)a$.

A *structure* over Γ is a mapping I assigning to every $a \in \Gamma$ a binary relation a^I . In this paper, we focus on finite structures, so the binary relations a^I must be finite unless explicitly specified otherwise. It is natural to view such a structure as a directed graph where edges are labeled by relation names. Accordingly we will refer to a pair (x, y) in a^I as an ‘ a -edge’ and denote it by $x \xrightarrow{a} y$.

The relation defined by an expression e in a structure I , denoted by $e(I)$, is defined inductively as follows:

- $a(I) = a^I$;
- $(e_1 \cup e_2)(I) = e_1(I) \cup e_2(I)$;
- $(e_1 \cap e_2)(I) = e_1(I) \cap e_2(I)$;
- $(e_1 - e_2)(I) = \{(x, y) \in e_1(I) \mid (x, y) \notin e_2(I)\}$;
- $(e_1 \cdot e_2)(I) = \{(x, y) \mid \exists z : (x, z) \in e_1(I) \text{ and } (z, y) \in e_2(I)\}$.

An expression e over Γ is called *finitely satisfiable* if there exists a structure I over Γ such that $e(I)$ is nonempty.

Remark 1. The standard notion of structure would include an explicit set U , called the domain of the structure, so that the relations a^I are binary relations on U . In the presence of a complementation operation this is important, as then the complement of a relation in a structure with domain U is taken with respect to $U \times U$. In our setting, however, we only have set difference, so an explicit domain would be irrelevant. Our notion of structure without an explicit domain actually agrees with the standard notion of ‘database instance’ in database theory [AHV95]. \square

Example 2. A trivial example of an unsatisfiable expression is $a - a$, but here is a less trivial example. For relation names a and b , the expression

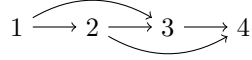
$$aaa - ((aa - b)a \cup ba)$$

is neither finitely satisfiable nor satisfiable by an infinite structure. In proof, consider a pair (x, y) that would belong to the result of evaluating this expression in some structure (for brevity we are omitting explicit reference to this structure). Then $(x, y) \in aaa$ so there exist a -edges (x, x_1) , (x_1, x_2) , and (x_2, y) . Since $(x, y) \notin (aa - b)a$, the b -edge (x, x_2) must be present. But then $(x, y) \in ba$, which is in contradiction with the last part of the expression. \square

Example 3. Expressions not involving the difference operator are always satisfiable, even by a finite series-parallel graph [DG06]. Using difference, we can give an expression e that is finitely satisfiable, but not by a series-parallel graph:

$$a(a \cap aa) - (aa - a)a$$

Indeed we have $(1, 4) \in e(W)$ where W is the canonical non-series-parallel graph [VTL82]:



To see that e cannot be satisfied by any series-parallel graph, suppose (x, y) belongs to the result of evaluating e on some structure. Since $(x, y) \in a(a \cap aa)$, there exist edges $x \rightarrow u_1 \rightarrow u_2 \rightarrow y$ and $u_1 \rightarrow y$ (we omit the labels on the edges which are all a). Since $(x, y) \notin (aa - a)a$, there must be an edge $x \rightarrow u_2$. If at least two of the four elements x, u_1, u_2 and y are identical, the graph contains a cycle and is not series-parallel. If all four elements are distinct, we have a subgraph isomorphic to W above, so the structure is not series-parallel [VTL82]. \square

Example 4. We can also give an example of an ‘infinity axiom’ in DA: an expression that is not finitely satisfiable but that is infinitely satisfiable. Let c be a third relation name apart from a and b , and consider the following expression e :

$$aba - (a(ba - a) \cup (a - ab)a \cup a(b - c)a \cup a(cc - c)a \cup a(cb \cap b)a)$$

To see that e is infinitely satisfiable, denote the set of natural numbers without zero by \mathbf{N} . Let ∞ denote an element that is neither zero nor in \mathbf{N} . Now consider the infinite structure I where

$$\begin{aligned} a^I &= \{(0, i) \mid i \in \mathbf{N} \ \& \ i \geq 2\} \cup \{(i, \infty) \mid i \in \mathbf{N}\} \\ b^I &= \{(i + 1, i) \mid i \in \mathbf{N}\} \\ c^I &= \{(j, i) \mid i, j \in \mathbf{N} \ \& \ j > i\} \end{aligned}$$

Then one can verify that $(0, \infty) \in e(I)$.

To see that e is not finitely satisfiable, suppose that (x, y) would belong to the relation defined by e in some finite structure. Then $(x, y) \in aba$ so there exist edges $x \xrightarrow{a} u_2 \xrightarrow{b} u_1 \xrightarrow{a} y$. Since $(x, y) \notin a(ba - a)$ we have also $u_2 \xrightarrow{a} y$.

Since $(x, y) \notin (a - ab)a$, there must exist edges $x \xrightarrow{a} u_3 \xrightarrow{b} u_2$. Again since $(x, y) \notin a(ba - a)$ we have also $u_3 \xrightarrow{a} y$. Continuing in this fashion we obtain an

infinite sequence u_1, u_2, \dots with edges $x \xrightarrow{a} u_i$ for every $i \geq 2$ and edges $u_i \xrightarrow{a} y$ and $u_{i+1} \xrightarrow{b} u_i$ for every $i \geq 1$.

Now since $(x, y) \notin a(b - c)a$ we have $u_{i+1} \xrightarrow{c} u_i$ for every $i \geq 1$. Then since $(x, y) \notin a(cc - c)a$ we have $u_j \xrightarrow{c} u_i$ for all $j > i \geq 1$. Since the structure is finite, there must exist $1 \leq i < j$ so that $u_i = u_j$. Hence we have a self-loop $u_j \xrightarrow{c} u_j$, implying $(x, y) \in a(cb \cap b)a$ which is in contradiction with the last part of the expression e . \square

The *finite satisfiability problem* for DA takes as input Γ and e , and asks to decide whether e is finitely satisfiable. We will show that this problem is undecidable already when Γ consists of a single relation name, and the difference degree of e is at most two. Here, the *difference degree*, denoted by $\deg e$, indicates how deeply applications of the difference operator are nested, and is inductively defined as follows:

- $\deg a = 0$;
- $\deg(e_1 \cup e_2) = \deg(e_1 \cap e_2) = \deg(e_1 \cdot e_2) = \max(\deg e_1, \deg e_2)$;
- $\deg(e_1 - e_2) = \max(\deg e_1, \deg e_2) + 1$.

The set of expressions with difference degree at most two is denoted by DA_2 . The set of DA_2 expressions over a single relation name is denoted by DA_2^1 . In Section 3, we will show that finite satisfiability for DA_2 is undecidable; in Section 4 we will show that this already holds for DA_2^1 .

Remark 5. Our focus on DA_2 explains why we have included intersection in DA, while this operator is actually redundant in the presence of difference by $r \cap s = r - (r - s)$. It appears that intersection is no longer redundant in DA_2 ; simulating it using difference would increase the difference degree by two times the number of nested applications of intersection. It remains open whether satisfiability of DA_2 expressions not using intersection is still undecidable.

3 Reduction from context-free grammar universality

Consider a context-free grammar $G = (\Sigma, V, S, P)$ with set of terminals Σ , set of nonterminals V , start symbol S , and set of productions P . Then G is called *universal* if $L(G)$, the language generated by G , equals Σ^* . Universality of context-free grammars is a well-known undecidable problem [HU79]. We will reduce the complementary problem, nonuniversality, to finite satisfiability of DA_2 expressions. The reduction will be based on a variation of the idea behind Example 4.

For technical reasons, we consider only grammars without empty productions, and redefine universality to mean that all nonempty strings over Σ belong to $L(G)$. Clearly, this notion of universality is still undecidable.

For any grammar G as above we construct a vocabulary Γ_G and a DA₂-expression e_G over Γ_G as follows. Choose three symbols α , ω and X not in $\Sigma \cup V$, and define $\Gamma_G = \Sigma \cup V \cup \{\alpha, \omega, X\}$. We define:

$$e_G = \varphi_0 - (\varphi_1 \cup \varphi_2 \cup \varphi_3 \cup \varphi_4 \cup \varphi_5 \cup \varphi_6 \cup \varphi_7),$$

where the subexpressions φ_i are defined as follows. We use Σ as a shorthand for $\bigcup_{b \in \Sigma} b$.

$$\begin{aligned}\varphi_0 &= \alpha \Sigma \omega \\ \varphi_1 &= \alpha \Sigma (\omega - \alpha) \\ \varphi_2 &= \alpha (\Sigma \alpha - \alpha) \\ \varphi_3 &= \bigcup_{Z_0 \rightarrow Z_1 \dots Z_n \in P} \alpha (Z_1 \dots Z_n - Z_0) \alpha \\ \varphi_4 &= (\alpha - \alpha \Sigma) S \omega \\ \varphi_5 &= \alpha (\Sigma - X) \alpha \\ \varphi_6 &= \alpha (X X - X) \alpha \\ \varphi_7 &= \alpha (X \Sigma \cap \Sigma) \alpha\end{aligned}$$

Proposition 6. *G is nonuniversal if and only if e_G is finitely satisfiable.*

Proof. The proof idea is an elaboration of the idea behind Example 4. For the only-if direction, assume there exists a nonempty word $b_1 \dots b_n$ not in $L(G)$. We must show that e_G is finitely satisfiable. Thereto we construct the following structure I over Γ_G :

$$\begin{aligned}\alpha^I &= \{(0, i) \mid i \in \{1, \dots, n\}\} \cup \{(i, \infty) \mid i \in \{1, \dots, n+1\}\} \\ \omega^I &= \{(n+1, \infty)\} \\ b^I &= \{(i, i+1) \mid i \in \{1, \dots, n\} \text{ \& } b_i = b\} \quad \text{for } b \in \Sigma \\ X^I &= \{(i, j) \mid i, j \in \{1, \dots, n\} \text{ \& } i < j\} \\ Y^I &= \{(i, j) \mid i, j \in \{1, \dots, n\} \text{ \& } i < j \text{ \& } b_i \dots b_{j-1} \in L(G, Y)\} \quad \text{for } Y \in V\end{aligned}$$

Here, $L(G, Y)$ is the set of words that can be generated from the nonterminal Y .

We claim that $(0, \infty) \in e_G(I)$. That $(0, \infty) \in \alpha \Sigma \alpha(I)$, and that

$$(0, \infty) \notin (\varphi_1 \cup \varphi_2 \cup \varphi_3 \cup \varphi_5 \cup \varphi_6 \cup \varphi_7)(I),$$

can be straightforwardly verified. To see that $(0, \infty) \notin \varphi_4(I)$, assume the contrary. Then there exist edges $0 \xrightarrow{\alpha} i \xrightarrow{S} j \xrightarrow{\omega} \infty$ in I so that $(0, i) \in (\alpha - \alpha \Sigma)(I)$. This is only possible for $i = 1$ and $j = n+1$. But then there is no edge $i \xrightarrow{S} j$ in I because $b_1 \dots b_n \notin L(G)$. Hence we have a contradiction.

For the converse direction, assume that G is universal. We show that e_G is not finitely satisfiable. It will be convenient to assume that G is in Chomsky

normal form [HU79], so that every production is of one of the two forms $Z_0 \rightarrow Z_1 Z_2$ or $Z_0 \rightarrow b$, with $Z_1, Z_2 \in V$ and $b \in \Sigma$.

Suppose, for the sake of contradiction, that some pair (x, y) belongs to the result of e_G evaluated in some finite structure I . To avoid clutter, in what follows we omit explicit references to I . Since $(x, y) \in \varphi_0$, there exist edges $x \xrightarrow{\alpha} u_2 \xrightarrow{b_1} u_1 \xrightarrow{\omega} y$ for some $b_1 \in \Sigma$. Since $(x, y) \notin \varphi_1$, we have also $u_1 \xrightarrow{\alpha} y$, and since $(x, y) \notin \varphi_2$, we have $u_2 \xrightarrow{\alpha} y$ as well. Since $(x, y) \notin \varphi_3$, we have $u_2 \xrightarrow{Y} u_1$ for every production $Y \rightarrow b_1$ in P .

The above construction of u_1 and u_2 forms the basis for the inductive construction of an infinite sequence u_1, u_2, \dots so that the following properties are satisfied for every natural number $n \geq 2$:

1. $u_1 \xrightarrow{\omega} y$;
2. $x \xrightarrow{\alpha} u_i$ for each $2 \leq i \leq n$, and $u_i \xrightarrow{\alpha} y$ for each $1 \leq i \leq n$;
3. for each $1 \leq i \leq n-1$ there is an edge $u_{i+1} \xrightarrow{b_i} u_i$ with $b_i \in \Sigma$;
4. for every $Y \in V$ and every $n \geq j > i \geq 1$ such that $b_{j-1} \dots b_i \in L(G, Y)$, there is an edge $u_j \xrightarrow{Y} u_i$.

Specifically, for any $m \geq 2$, assume we already have defined u_1, \dots, u_m ; we then define u_{m+1} as follows. Since G is universal, $b_{m-1} \dots b_1 \in L(G)$. Hence, by property (4) above, $u_m \xrightarrow{S} u_1$. Since $(x, y) \notin \varphi_4$, there must exist an element u with edges $x \xrightarrow{\alpha} u \xrightarrow{b_m} u_m$ for some $b_m \in \Sigma$. We set $u_{m+1} := u$ and check that the above properties are still satisfied.

For property (1) nothing has changed. For property (2) we have $x \xrightarrow{\alpha} u_{m+1}$ given, and $u_{m+1} \xrightarrow{\alpha} y$ follows from $(x, y) \notin \varphi_2$. For property (3), we have $u_{m+1} \xrightarrow{b_m} u_m$ given. For property 4, we verify this by induction on the length of the string $b_{j-1} \dots b_i$. If $j = i+1$, the production $Y \rightarrow b_i$ belongs to P and we have $u_j \xrightarrow{Y} u_i$ by $(x, y) \notin \varphi_3$. If $j > i+1$, consider a derivation tree of $b_{j-1} \dots b_i$ from Y , and let $Y \rightarrow Z_1 Z_2$ be the production used at the root of the derivation tree. Then there exists k strictly between j and i so that $b_{j-1} \dots b_k \in L(G, Z_1)$ and $b_{k-1} \dots b_i \in L(G, Z_2)$. By induction we have edges $u_j \xrightarrow{Z_1} u_k \xrightarrow{Z_2} u_i$, which implies $u_j \xrightarrow{Y} u_i$ by $(x, y) \notin \varphi_3$.

Now since $(x, y) \notin \varphi_5$, we have $u_{i+1} \xrightarrow{X} u_i$ for each $i \geq 1$. Then since $(x, y) \notin \varphi_6$, we have $u_j \xrightarrow{X} u_i$ for all $j > i \geq 1$. Since the structure is finite, there must exist $1 \leq i < j$ so that $u_i = u_j$. Hence we have a self-loop $u_j \xrightarrow{X} u_j$, implying $(x, y) \in \varphi_7$ which is in contradiction with $(x, y) \in e_G$. \square

4 Reduction to a single relation name

In this section we establish our main theorem:

Theorem 7. *The finite satisfiability problem for DA_2^1 is undecidable.*

The result of the previous Section already implies the undecidability of the finite satisfiability problem for DA_2 . Hence, to prove the above Theorem, it suffices to translate any given expression e over any given vocabulary Γ to an expression e' over a single relation name, so that $\deg e = \deg e'$ and e is satisfiable if and only if e' is.

We will do this in two steps. In a first step, we will reduce to two relation names; in the second step we reduce further from two to one.

Let $\Gamma = \{a_1, \dots, a_k\}$ ordered in an arbitrary manner and let b and c be two symbols not in Γ . We define e' as the expression obtained from e by replacing every occurrence of a_i , for $i = 1, \dots, k$, by $b(c \cap c^{i+1})b$, where c^j denotes the composition $c \cdots c$ (j times).

Proposition 8. *e is finitely satisfiable if and only if e' is.*

Proof. For the if-direction, we convert any structure J over $\{b, c\}$ to a structure K over Γ as follows: for each $a_i \in \Gamma$, we set $a_i^K = b(c \cap c^{i+1})b(J)$. It is now readily verified by structural induction that $e'(J) = e(K)$ for every expression e . In particular, if $e'(J)$ is nonempty, then so is $e(K)$.

For the only-if direction, we convert any structure K over Γ to a structure J over $\{b, c\}$ as follows. Recall [AHV95] that the *active domain* of K , denoted by $\text{adom}(K)$, equals the set of all elements that appear as first or second component of a pair in a relation of K . Now for each $i = 1, \dots, k$ and each $(x, y) \in a_i^K$, choose a set $\{u_1^{x,y,i}, \dots, u_{i+2}^{x,y,i}\}$ of $i+2$ distinct elements. All these sets must be pairwise disjoint and disjoint from $\text{adom}(K)$. Then b^J consists of all edges $x \rightarrow u_1^{x,y,i}$ and $u_{i+2}^{x,y,i} \rightarrow y$ for every $i = 1, \dots, k$ and every $(x, y) \in a_i^K$. Moreover c^J consists of all edges

$$u_1^{x,y,i} \rightarrow \dots \rightarrow u_{i+2}^{x,y,i} \text{ and } u_1^{x,y,i} \rightarrow u_{i+2}^{x,y,i}$$

for every $i = 1, \dots, k$ and every $(x, y) \in a_i^K$.

For every expression e we now again claim that $e(K) = e'(J)$. We can prove this again by induction on the structure of e . The only potential difficulty is present in the basis of the induction, where e is a relation name $a_i \in \Gamma$. The inclusion $e(K) \subseteq e'(J)$ holds by construction. For the converse inclusion, assume $(u, v) \in b(c \cap c^{i+1})b(J)$. Then there exist edges $u \xrightarrow{b} z_1 \xrightarrow{c} z_2 \xrightarrow{b} v$ such that $(z_1, z_2) \in c^{i+1}(J)$. Due to the edge $u \xrightarrow{b} z_1$, there are only two possibilities for u :

- u equals $u_{j+2}^{x,y,j}$, for some x, y and j such that $(x, y) \in a_j^K$. Then z_1 must be y . However, by $z_1 \xrightarrow{c} z_2$, this is impossible, since there is no c -edge leaving y .
- u equals x , for some y and j such that $(x, y) \in a_j^K$. Then z_1 is $u_1^{x,y,j}$ and there are two possibilities for z_2 :

1. z_2 is $u_2^{x,y,j}$. By $z_2 \xrightarrow{b} v$ this is impossible, since there is no b -edge leaving $u_2^{x,y,j}$.
2. z_2 is $u_{j+2}^{x,y,j}$, so v is y . Since $(z_1, z_2) \in c^{i+1}(J)$, and the only chain of c -edges from $u_1^{x,y,j}$ to $u_{j+2}^{x,y,j}$ is the chain $u_1^{x,y,j} \rightarrow \dots \rightarrow u_{j+2}^{x,y,j}$, we must have $j = i$. Hence, we obtain that $(u, v) = (x, y) \in a_i^K$ as desired.

□

For the reduction to a single relation name, consider any expression e over the vocabulary $\{b, c\}$ with two relation names, and let a be a third symbol. We define the expression \hat{e} over the vocabulary $\{a\}$ as the expression obtained from e by replacing every occurrence of b by $a(a \cap a^2)a$ and every occurrence of c by $a(a \cap a^3)a$. Again we show:

Proposition 9. *e is finitely satisfiable if and only if \hat{e} is.*

Proof. For the if-direction, we convert any structure J over $\{a\}$ to a structure I over $\{b, c\}$ as follows: $b^I = a(a \cap a^2)a(J)$ and $c^I = a(a \cap a^3)a(J)$. It is now readily verified by structural induction that $\hat{e}(J) = e(I)$ for every expression e over $\{b, c\}$. In particular, if $\hat{e}(J)$ is nonempty, then so is $e(I)$.

For the only-if direction, we convert any structure I over $\{b, c\}$ to a structure J over $\{a\}$ as follows. For every edge $x \xrightarrow{b} y$ in I we choose a set $\{u_1^{x,y,b}, \dots, u_3^{x,y,b}\}$ of three distinct elements; for every edge $x \xrightarrow{c} y$ in I we choose a set $\{u_1^{x,y,c}, \dots, u_4^{x,y,c}\}$ of four distinct elements. All these sets must be pairwise disjoint and disjoint from $\text{adom}(I)$. We now define a^J to consist of all edges

$$x \rightarrow u_1^{x,y,b} \rightarrow u_2^{x,y,b} \rightarrow u_3^{x,y,b} \rightarrow y \text{ and } u_1^{x,y,b} \rightarrow u_3^{x,y,b}$$

for every edge $x \xrightarrow{b} y$ in I , plus all edges

$$x \rightarrow u_1^{x,y,b} \rightarrow u_2^{x,y,b} \rightarrow u_3^{x,y,b} \rightarrow u_4^{x,y,b} \rightarrow y \text{ and } u_1^{x,y,b} \rightarrow u_4^{x,y,b}$$

for every edge $x \xrightarrow{c} y$ in I .

We now make Claim B and Claim C.

Claim B: $b^I = a(a \cap a^2)a(J)$. The inclusion from left to right holds by construction. For the inclusion from right to left, let $(u, v) \in a(a \cap a^2)a(J)$. Then there exist edges $u \rightarrow z_1 \rightarrow z_2 \rightarrow v$ in J with $(z_1, z_2) \in (a \cap a^2)(J)$. An obvious possibility is that $z_1 = u_1^{x,y,b}$ and $z_2 = u_3^{x,y,b}$ for some $(x, y) \in b^I$. Then u must equal x and v must equal y so $(u, v) = (x, y) \in b^I$ as desired. Let us now verify that there are no other possibilities for z_1 and z_2 . Thereto we list all other possibilities for a pair $(z_1, z_2) \in a^2(J)$:

- $(u_1^{x,y,b}, y)$ with x and y as above;
- $(u_2^{x,y,b}, y)$;

- $(x, u_2^{x,y,b});$
- $(x, u_3^{x,y,b});$
- $(u_3^{x,y,b}, u_1^{y,z,r}),$ with $r = b$ or c , for some z such that $(y, z) \in r^I;$
- $(u_1^{x',y',c}, y')$ for some $(x', y') \in c^I;$
- $(u_1^{x',y',c}, u_3^{x',y',c});$
- $(u_2^{x',y',c}, u_4^{x',y',c});$
- $(u_3^{x',y',c}, y');$
- $(u_4^{x',y',c}, u_1^{y',z',r}),$ with $r = b$ or c , for some z' such that $(y', z') \in r^I;$
- $(x', u_2^{x',y',c});$
- $(x', u_4^{x',y',c}).$

In all these cases, there is no edge $z_1 \rightarrow z_2$ in J , so that $(z_1, z_2) \notin (a \cap a^2)(J)$.

Claim C: $c^I = a(a \cap a^3)a(J)$. The inclusion from left to right holds by construction. For the inclusion from right to left, let $(u, v) \in a(a \cap a^3)a(J)$. Then there exist edges $u \rightarrow z_1 \rightarrow z_2 \rightarrow v$ in J with $(z_1, z_2) \in (a \cap a^3)(J)$. The obvious possibility is that $z_1 = u_1^{x,y,c}$ and $z_2 = u_4^{x,y,c}$ for some $(x, y) \in c^I$. Then u must equal x and v must equal y so $(u, v) = (x, y) \in c^I$ as desired. We now verify that there are no other possibilities for z_1 and z_2 . Thereto we list all other possibilities for a pair $(z_1, z_2) \in a^3(J)$:

- $(u_1^{x,y,c}, u_1^{y,z,r}),$ with x and y as above, $r = b$ or c , and some z such that $(y, z) \in r^I;$
- $(u_2^{x,y,c}, y);$
- $(u_3^{x,y,c}, u_1^{y,z,r});$
- $(u_4^{x,y,c}, u_2^{y,z,r});$
- $(u_4^{x,y,c}, u_3^{y,z,r})$ if $r = b;$
- $(u_4^{x,y,c}, u_4^{y,z,r})$ if $r = c;$
- $(x, u_3^{x,y,c});$
- $(x, y);$
- $(u_1^{x',y',b}, y')$ for some $(x', y') \in b^I;$
- $(u_1^{x',y',b}, u_1^{y',z',r'}),$ with $r' = b$ or c , for some z' such that $(y', z') \in r'^I;$
- $(u_2^{x',y',b}, u_1^{y',z',r'});$
- $(u_3^{x',y',b}, u_2^{y',z',r'});$
- $(u_3^{x',y',b}, u_3^{y',z',r'})$ if $r' = b;$
- $(u_3^{x',y',b}, u_4^{y',z',r'})$ if $r' = c;$

- $(x', u_3^{x', y', b});$
- $(x', y').$

In all these cases, there is no edge $z_1 \rightarrow z_2$ in J , so that $(z_1, z_2) \notin (a \cap a^3)(J)$.

From Claims B and C it now follows readily by structural induction that $e(I) = \hat{e}(J)$ for every expression e over $\{b, c\}$. In particular, if $e(I)$ is nonempty, then so is $\hat{e}(J)$. \square

5 Conclusion

In DA_2 -expressions, applications of the set difference operation can be nested at most once. It is thus natural to wonder what happens in the fragment where set difference cannot be nested at all. In a companion paper, we consider the fragment of the full Tarski Algebra (TA), with general complementation, defined by the restriction that complement can only be applied to expressions that do not already contain an application of complement. It turns out that finite satisfiability for TA-expressions without nested complement is decidable and even belongs to NP.

As already mentioned in Remark 5, it remains open whether satisfiability for DA_2 -expressions without the intersection operation is decidable. As already mentioned in the Introduction, the decidability of unrestricted satisfiability for DA remains open as well.

Acknowledgment

We are indebted to Stijn Vansummen for a number of inspiring discussions on the topic of this paper.

References

- [AGN97] H. Andréka, S. Givant, and I. Németi. *Decision problems for equational theories of relational algebras*, volume 126 of *Memoirs*. AMS, 1997.
- [AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [BCM⁺03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [BGG97] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer, 1997.

- [DG06] D.J. Dougherty and C. Gutierrez. Normal forms for binary relations. *Theoretical Computer Science*, 360(1–3):228–246, 2006.
- [FGL⁺11] G.H.L. Fletcher, M. Gyssens, D. Leinders, J. Van den Bussche, D. Van Gucht, S. Vansummen, and Y. Wu. Relative expressive power of navigational querying on graphs. In *Proceedings 14th International Conference on Database Theory*, 2011.
- [Fig12] D. Figueira. Decidability of downward XPath. *ACM Transactions on Computational Logic*, 13(4):article 34, 2012.
- [HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [KRV14] E.V. Kostylev, J.L. Reutter, and D. Vrgoč. Containment of data graph queries. In *Proceedings 17th International Conference on Database Theory*. ACM, 2014.
- [LMV13] L. Libkin, W. Martens, and D. Vrgoč. Querying graph databases with XPath. In *Proceedings 16th International Conference on Database Theory*. ACM, 2013.
- [LW05] C. Lutz and D. Walther. PDL with negation of atomic programs. *Journal of Applied Non-Classical Logics*, 15(2):189–213, 2005.
- [Mad91] R.D. Maddux. The origin of relation algebras in the development and axiomatization of the calculus of relations. *Studia Logica*, 50(3/4):421–455, 1991.
- [MV97] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*. Springer, 1997.
- [Pra92] V. Pratt. Origins of the calculus of binary relations. In *Proceedings 7th Annual IEEE Symposium on Logic in Computer Science*, pages 248–254, 1992.
- [Sch79] W. Schönfeld. An undecidability result for relation algebras. *Journal of Symbolic Logic*, 44(1):111–115, 1979.
- [SSVG93] V.M. Sarathy, L.V. Saxton, and D. Van Gucht. Algebraic foundation and optimization for object based query languages. In *Proceedings 9th International Conference on Data Engineering*, pages 81–90. IEEE Computer Society, 1993.
- [Tar41] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6:73–89, 1941.

- [tCM07] B. ten Cate and M. Marx. Navigational XPath: Calculus and algebra. *SIGMOD Record*, 36(2):19–26, 2007.
- [TG87] A. Tarski and S. Givant. *A Formalization of Set Theory Without Variables*, volume 41 of *AMS Colloquium Publications*. American Mathematical Society, 1987.
- [VTL82] J. Valdes, R.E. Tarjan, and E.L. Lawler. The recognition of series parallel digraphs. *SIAM Journal on Computing*, 11:298–313, 1982.
- [Woo12] P. Wood. Query languages for graph databases. *SIGMOD Record*, 41(1):50–60, March 2012.